REMARKS

The examiner has rejected the independent claim (62) under 35 USC 103(a) as being unpatentable over Tolpin combined with Nakagawa. The examiner is urged to reconsider and withdraw the rejection.

The claims have been amended to better describe the invention. Claim 62 now reads:

> 62. A method by which a disk-based distributed data storage system is organized for protecting historical records of stored data entities, the method comprising:
>
> recording distinct states of stored data entities, corresponding to different moments of time, as a plurality of entity versions coexisting within the distributed data storage system;
>
> storing copies of an entity version that is one of the plurality of entity versions at each of a plurality of storage sites of the distributed data storage system;
>
> sharing, among the plurality of storage sites, a set of rules that restrict deletion of the entity versions; and
>
> applying the shared set of rules independently at each of the plurality of storage sites, to determine whether or not the copies of the entity version can be deleted;
>
> wherein if it is determined that the copies of the entity version cannot be deleted then they also cannot be modified;
>
> wherein deletion of the copies of the entity version from the distributed data storage system is only allowed if the set of rules determine that the entity version can be deleted; and
>
> wherein an action taken by a client program communicating with the disk-based distributed data storage system causes the shared set of rules to restrict deletion of the entity version and no subsequent action taken by the client program can cause deletion to occur in violation of the restriction.

The claim describes a method for protecting historical records of stored data entities ("entity versions") on a distributed data storage system. Copies of entity versions are stored across a plurality of storage sites. A set of rules that restrict deletion of entity versions is shared among the storage sites. The rules are applied independently at the storage sites, to determine whether the copies of the entity versions may be deleted at that site (if they cannot be deleted,

they can also not be modified). Deletion is governed by the rules, and so copies of the entity versions can only be deleted if the rules permit such action. Once a client program has communicated with the data storage system so as to cause the rules to restrict deletion, no subsequent action by the client program can cause deletion to occur in violation of the rules.

Neither Tolpin nor Nakagawa, nor any other art of record, comes even close to suggesting the invention of claim 62. Neither Tolpin nor Nakagawa deals with a distributed storage system, let alone with sharing a set of rules across a distributed system and having the rules independently applied at the different storage sites. The examiner has misunderstood the references in concluding otherwise. E.g., the examiner refers to paragraph 26 of Tolpin for support for sharing a set of rules among a plurality of storage sites, but the paragraph says nothing of the kind. Here is the full text of the paragraph:

> [0026] The value of the preservation weight is preferably within a predetermined range of possible preservation weights. Preferably, the lower the preservation weight, the less important it is that the member, and thus the data of the member, be preserved by the system for any extended period of time. Likewise, the higher the preservation weight, the greater the importance of preservation of the member's data. In one embodiment, the highest preservation weight indicates that the data should be kept in the finite data storage permanently (i.e., it should never be deleted, if possible). Obviously, a system that merely reverses the scale such that the lowest preservation weight indicates the greater importance of the data, and vice versa, is an identical priority system and within the scope of the present invention. The specific value of the preservation weight may be assigned by default, by a system administrator, or by a user with access to the underlying computer configuration and with rights to assign preservation weights to particular snapshots or backups. The preservation weight may also be automatically determined in accordance with a predetermined function, which may be based on the time of creation of the member, the type of information represented by the data of the member, or the type of user for or by which the member was created. Additionally, preferably a particular preservation weight assigned to a member may be changed, for example, by a user or administrator after data thereof is stored in the finite data storage. In this scenario, all members are sorted prior to performing any deletion each time the consumption level of the finite data storage reaches or exceeds a trigger level for initiating the deletion subroutine.

It is clear from other portions of the reference that Tolpin deals with a single storage site, rather than a distributed storage system. For example, paragraph 22, refers to a single storage medium:

> [0022] Turning first to FIG. 1, a computer configuration 100 for implementing preferred methods of the present invention is illustrated. This computer configuration 100 comprises a storage medium 110. The storage medium 110 is a data storage area of the computer configuration 100 from which data is accessible. The storage medium 110 thus may include, for example, the entirety or any portion of a hard disk drive, computer RAM-memory, a USB data storage device, a serial data storage device, a parallel data storage device, or firewire data storage device. As shown in FIG. 1, a finite data storage 112 comprising a fixed percentage or portion of the storage medium 110 is allocated to preserving data of backups and/or snapshots. The remaining storage area 114 of the storage medium 110 stores primary or current data. The system 100 further comprises a central processor 120, which contains the operating system of the computer configuration 100 and which interacts with other peripheral devices (not shown) of a conventional computer configuration, such as user or data interfaces (e.g. terminals, monitors, keyboards, mouse(s), scanners), output devices (e.g., printers), and I/O devices (e.g. modems). The central processor 120 interacts with the storage medium 110 by means of a data manager 130. The data manager 130 comprises software and/or hardware for managing the transfer of data between the processor 120 and the storage medium 110 and for controlling the writing, reading, and deletion of data to and from the storage medium 110. A preferred data manager is disclosed utilized by the source code set forth in the provisional patent application, previously incorporated herein by reference.

Nakagawa also has nothing to do with a distributed storage system, let alone with rules shared among storage sites. It is concerned, instead, with how to organize folders within a storage system.

Accordingly, claim 62 is allowable over the art of record.

The remaining claims are all properly dependent on claim 62, and thus allowable therewith. Each of the dependent claims adds one or more further limitations that enhance patentability, but those limitations are not presently relied upon. For that reason, and not because applicants agree with the examiner, no rebuttal is offered to the examiner's reasons for rejecting the dependent claims.

Allowance of the application is requested.

Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 1/16/2007

/grogerlee/
G. Roger Lee
Reg. No. 28,963

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906